Reply to the referee 2

October 18, 2023

We would like to thank the referee for the feedback and suggestions to improve the article. The technical comments (or typos) have been corrected directly in the manuscript, while the questions that require a more detailed explanation have been answered in this document, also modifying accordingly the respective part in the manuscript. Besides, some of the comments were already addressed since they had been already pointed out by the referee 1. We encourage the referee 2 to take a look at the response to the previous referee if the answer provided here seems too short. The answers are indicated in blue throughout the whole document.

1. Why are approaches using polygons or penalty functions not sufficient, in the opinion of the authors? A few details would be welcome here.

   In the literature review, we observed that commonly the concave shapes involve a challenge when using polygons, and this is something that our formulation can deal with in a robust way. Regarding the penalty functions, as we also mention in our answer to the referee 1, we are aware that it could be applied to the problem, but we just simply decided to go for this alternative. Besides, as also mentioned in our reply, applying a penalty function would require some type of smoothing term to avoid turbines oscillating in and out of the exclusion zones.

2. "If more than one polygon of the same type overlap, these are merged into a single one". This is fine, in principle. But what about inclusion and exclusion zones - I assume that these are not allowed to overlap?

   Both types are allowed to overlap. This is a very nice feature of the method, as we can first define all the polygons, assigning the inclusion or exclusion attribute to each of them, and then superpose all of them. The resulting shapes are the merged shapes created by the interception of all shapes. If two shapes do not intercept, but overlap, then depending on the case there are two options. To illustrate this let's imagine a big circle area and a second circle with the same center but smaller radius. If the larger circle is an inclusion zone and the smaller circle is an exclusion zone, then the resulting inclusion zone from the merge will be a ring; in the opposite case, the resulting inclusion zone from the merge will be the smaller circle.

3. What is a "unitary" vector? I assume the authors mean that these vectors are normalized to unit length?

   That is correct. We have added the next line to the manuscript in order to make it more clear: *"(...) we define a normal unitary vector (a vector whose module is equal to the unit of length) that points inside the inclusion zone polygons and outside the exclusion zone polygons (...)"*.

4. It is very confusing to see the normal vectors drawn at the starting node of their respective edge, instead of - as I would expect - somewhere around the middle of the edge. Mathematically, there is no problem, but it goes somehow against expectations of how to visualize these things. Please reconsider.

   This is a fair point. We have corrected the Figure in the manuscript with the normal vectors drawn in the middle of the edges.

5. I would assume that this "greedy" algorithm could also fail under certain conditions? Moreover, since it seems to be deterministic (when r = 0), there would then be no starting condition for the actual optimization? Is this the (main) reason for adding randomness to this procedure? (I think there are

straightforward ways to deal with this problem better, though - maybe something the authors might want to think about)

This question requires an answer of several parts. For the first part: yes, the algorithm can fail under certain conditions, and one of those conditions is that there are not enough points to satisfy the constraints. For example, let's imagine a square with 5 rotor diameters of side length. If we want to allocate 4 wind turbines with a spacing constraint of 5 rotor diameters between each turbine, they would only fit in the square if they are located on the corners. Therefore, there is a feasible solution but the algorithm would fail if it places one of the turbines in a different position than a corner.

About the algorithm being deterministic, we need to clarify that it would be deterministic for this case, without using randomness, because the wind resource is not uniform, but it won't be deterministic for a uniform resource site since the first wind turbine will be always random and that would affect the final result. The randomness might still be useful with a non-uniform resource in certain cases, for instance, let's imagine a case with 2 polygons and non-uniform wind resource. One of this two polygons has much more resource than the other, but very limited space, whereas the other polygon has plenty of space to locate turbines. Now let's also imagine that the polygon with higher potential has a shape that, due to minimum spacing constraints between turbines, if one of them is put in the middle, then there won't be space for additional turbines within, whereas if we put the turbine in one of the corners, there will be space for another turbine. In a case like this, placing two wind turbines in the polygon with higher resource would lead to higher AEP than placing only one of them, and if the middle of the polygon would be slightly better, the smart-start with 0 randomness would place a turbine in the middle, not reaching the best solution.

6. Eq. 15: I assume the gradients of the velocity deficit are also needed, when determining the gradient of the AEP? But this is handled by algorithmic differentiation, correct?

   Yes, we use the python library autograd to differentiate the code automatically, so we did not include this expression in the manuscript.

7. line 314ff: "there might not be enough time (iterations) to explore the domain". Well, the relaxation of the distance by way of Eq. 9 prescribes a specific way in which this constraint is changing from iteration to iteration. In some cases the optimization even terminates when the distance function is still relaxed (line 361), which the authors consider to be invalid optimization runs!? The first seems somewhat ad-hoc, and the second seems somewhat strange. I would imagine that it would be much more natural to fully optimize the wind farm layout for a fixed offset of the distance, and only then adjust the constraint and solve the next (slightly less relaxed) optimization problem - using the previous optimal solution as an initial (though most likely infeasible) state. Thus, instead of considering one optimization problem with continuously changing constraint relaxation this would be a series of optimization problems with different (but fixed) constraint relaxations - until the final problem with no relaxation is solved. Why did the authors not try this, very often used, strategy?

   The only reason why we discard the optimization runs in which the solver terminates before the relaxation finishes (or better said, before the un-relaxation finishes), is due to the fact that some of the wind turbines might find the sweet spot out of the true boundaries. That means that this solution is unfeasible. The approach described by the referee (allow for the optimization to find the optimum and then restart it with smaller bounds) makes total sense, but it would take more time, and hence we would lose one of the desired features to use gradient-based optimization: the speed.

8. Continuing from the previous comment, there might also be better ways to change the relaxation offset to the distance adaptively, i.e., depending on the speed of convergence (how much better the solution becomes between iterations).

   We agree that the relaxation function is somehow very basic and could be improved. Initially, the scope of the paper was to demonstrate the method to include polygons as boundaries while using the gradient-based method, so we did not explore very much the posibilities that could be applied both to the relaxation function or to the smart-start algorithm. This is rather something that could be considered for future work.

9. line 321: "before relaxation is applied to the last 300 meters." - There seems to be some confusion about the use of the term "relaxation" and I would like the authors to review their use of it. The relaxed optimization problem has the least strict constraint. It is thus not correct to talk about relaxing the constraint in later iterations where the constraint actually becomes more strict! Similar issues of imprecise use occur in line 346 ("no clear benefit in slowing down the relaxation") and line 375 ("high speed of the relaxation").

   This point was also made by the first referee, and we adapted the manuscript to his suggestion, changing the term 'relaxation' by 'un-relaxation' when it proceeds. We totally agree with this and we hope that the main text now has become more understandable for the reader.

10. line 325: Why the value of 97? Why not 103? There seems to be some inconsistency between this here and Eq. 17.

    This was a numerical error. It has been corrected in the main text.

11. Table 2: A comment seems appropriate why not both relaxation and smart-start have been tested together.

    The combination of the relaxation technique and the *smart-start* would place the turbines within infeasible areas and then, in many cases, the optimization would converge so fast that we would end in one of the situations in which we discarded those seeds. In any case, we do not believe that it would get better solutions that the single use of the *smart-start* combined with SLSQP. We did not explain this in the main manuscript because we think it might lead to confusion.

12. page 17. The results are discussed mainly in terms of average AEP, but what about the maximum values? These are, after all, what one is looking for when trying to optimize the problem!

    That is a fair point. Since the article is focused on the methodology, we decided to analyze the results from the statistical perspective, but in practice, it seems more reasonable to focus on the solutions that lead to highest AEP. However, it can be seen from the violin plots that the maximum solutions reached by the relaxation approach and the *smart-start* are very close.

13. Fig. 9: I am somewhat surprised to see so much variability in the results, depending on the used random seed / initial position of the turbines. Obviously there seem to be many local maxima in this problem. Although the approach of the authors is promising, this seems still to be a major problem - how to find the actual, global optimum? Some thoughts along this direction would be highly appreciated in the discussion.

    Indeed, it is a multimodal optimization problem with a lot of constraints that is solved with a gradient-based solver. This involves that the found optimum will be sensitive to the initial guess. On top of that, the combination of wind turbines per polygon also plays an important role in finding the global optimum. Unfortunately, it would be challenging to ensure that the global optimum has been found, as we are working on the continuous space and it becomes impractical to exhaustively explore all the possibilities.

14. Coming back to the problem of local maxima, have the authors considered to use the idea of marginal improvements - i.e., estimating numerically how much the addition or removal of a single wind turbine from each inclusion zone would add / remove from the overall AEP (per turbine)? This type of sensitivity could then be used to determine when or how to move turbines between inclusion zones...

    This approach is very interesting and we did not considered it. When thinking about how to put it into practice, there is the additional problem of deciding where to put the added / removed wind turbine, and given that the domain has variable wind resource, it becomes an optimization problem within another optimization problem (nested problem). We imagine this as a time consuming problem but will be kept in mind for future development.